Modular incremental modelling of belief and intention

Bunt, H.C.

Publication date:
1990

Document Version
Publisher's PDF, also known as Version of record

TILBURG UNIVERSITY
KATHOLIEKE UNIVERSITEIT BRABANT

[redacted]

ITK RESEARCH REPORT

INSTITUTE FOR LANGUAGE TECHNOLOGY AND ARTIFICIAL INTELLIGENCE

INSTITUUT VOOR TAAL- EN KENNISTECHNOLOGIE

[redacted]

ITK Research Report No. 20

July 27, 1990

Modular Incremental Modelling of Belief and Intention

Harry Bunt

Paper, presented at the Second International Conference on User Modelling, Honolulu, March-April 1990.

ISSN 0924-7807

Modular Incremental Modelling of Belief and Intention

Harry Bunt
Institute for Language Technology and Artificial Intelligence
Tilburg, The Netherlands
Bunt@kub.nl

Abstract

This paper decribes a formalism for modular, incremental modelling of the beliefs and intentions of the human user of an intelligent interactive information system.

The formalism makes use of partial models of a formal logical language with provisions for expressing epistemic and intentional attitudes, the choice of which is inspired by an analysis of the transfer of information in natural-language information-seeking dialogues. These models are designed to be incremental in the sense that the model grows as more information becomes available; this in contrast with standard logical models for expressing knowledge or belief.

The formalism is modular in that information relating to different epistemic and intentional attitudes is contained in different 'modules', with the result that only certain modules have to be consulted in order to decide whether the model represents certain information, and that only certain modules have to be considered when information is to be stored in the model.

The use of the model in dialogue management is briefly considered. The construction and update of a model on the basis of interpreting incoming communicative actions is described, and the generation of communicative acts on the basis of the model is outlined.

1 Intelligent dialogue and user modeling

Intelligent communicative behaviour is the performance of communicative actions in compliance with one's goals and information state, in particular with the available information about the partner's goals, beliefs, and information state.

This applies both to human-computer communication and to the exchange of messages between cooperating processors in distributed computing (see Bunt, 1989 and Halpern & Moses, 1984, respectively). In the case of human-computer communication, the "information state" required in a dialogue system includes a model of what the user wants, knows, believes, does not know, wants to know, believes that the system knows, etc. In other words, it includes a *user model.*

In order to have a meaningful interaction with a human partner about a certain domain of discourse, a dialogue system of course needs, in addition, to have domain knowledge as another part of its information state. For many applications, this knowledge can be considered fixed; the user model, by contrast, is inherently dynamic: the user's communicative actions constantly provide the system with new information about the user's goals, intentions and information. The processing of this new information, i.e. the way the user model is updated, and the generation of responses on the basis of the updates in the user model, is what I refer to as "dialogue management". It forms the "engine" of intelligent communication, and it is a process which by its very conception involves a user model.

2 The analysis of communicative action

Of central importance for intelligent human-computer communication is obviously that the machine *understands* the user. To understand a communicative action means to know what information the action conveys about the user's intentions, beliefs, expectations, hopes, etc. In this paper we restrict ourselves to the kind of communication which we call an "information dialogue", i.e. a dialogue with the sole purpose of exchanging factual information. This has the advantage that the attitudes towards information that we have to take into account in user models are restricted to epistemic ones: knowing something, wanting to know something, etc.

An information dialogue is an exchange of communicative acts between two partners S and U with the purpose of obtaining or providing factual information. The communicative acts that S and U perform serve this purpose: they provide factual information, request factual information, verify factual information, (dis)confirm factual information, etc. Not all communicative acts are about factual information, however. Sometimes a dialogue participant refers explicitly to himselfor the other, or to the way the dialogue is developing; such communicative acts are called ***dialogue control acts*** (Bunt, 1986). The following example illustrates this.

(1)	1	S:	Amsterdam Airport information
	2	*U*:	Good afternoon, this is Van I. in Eindhoven. I would like to have some information about flights to Munich. When can I fly there between now and ... next Sunday
	3	S:	Let me have a look. Just a moment
	4	U:	Yes
	5	S:	O.K., there are ... three flights every day, one at nine fifty,
	6	U:	Yes,
	7	S:	one at one-forty ... and one at six twenty-five
	8	U:	Six twenty-five ... These all go to Munich
	9	S:	These all go to Munich
	10	U:	And that's on Saturday too
	11	S:	And that's on Saturday too, yes
	12	U:	Right ... Do you also have information about the connections to Schiphol by train?
	13	S:	Yes, I do.
	14	U:	Do you know how long the train ride takes to Schiphol?
	15	S:	You are travelling from Eindhoven?
	16	U:	That's right.
	17	S:	It's nearly two hours to Amsterdam ... You change there and two and a half hours
	18	U:	O.K., thank you
	19	S:	You're welcome
	20	U:	Bye
	21	S:	Bye

Sheer factual information is at stake in this dialogue only in the last communicative act of turn 2, and in the turns 5, 7, 8, 9, 10, 11, 14 and 17; the remaining acts, in the turns 1, 2, 3, 4, 6, 8, 12, 13, 15, 16, 18, 19, 20, and 21 are all dialogue control acts. They serve, roughly speaking, to establish contact (1 and 2), to maintain contact (3 and 4; 12), to acknowledge reception

(6 and 8; 18), to verify (8 and 10; 15 and 16), to investigate availablity of information (12 and 13), to reach agreement on closing the dialogue (18 and 19) and to close the dialogue (20 and 21). This illustrates our finding that in information dialogues in a wide variety of conditions roughly speaking half of the communicative acts are dialogue control acts (Bunt, 1986).

The example dialogue also illustrates that one and the same sentence may be used with different purposes, conveying correspondingly different information. The sentence *These all go to Munich* as used in turn 8 by U, has on S (among other things) the effect that S knows that U wants to know whether the flights in question all go to Munich, whereas the same sentence used subsequently by S has the effect on U that U now knows that indeed these all go to Munich. These differences are due to the fact that the first use of the sentence has the function of a question (more precisely, of a *check*; it is an example of a so-called *declarative question act*; see Beun, 1989), whereas the second use has the function of an answer (more precisely, of a *confirmation*). The relevant units in a dialogue are therefore not sentences, but *utterances*: sentences used with a certain communicative function, or conceived at a more abstract level, *communicative acts:* combinations of communicative function and semantic content.

It is, evidently, of fundamental importance to the understanding of an utterance in an information dialogue to determine what information the speaker has available and in what respects he wants to expand his information. Instead of saying that S has certain information available, we sometimes say that S 'knows' something. But we must be careful using the term 'knowledge', for two reasons. First, we should perhaps speak of 'belief' rather than 'knowledge', in order to avoid the implication that the available information is necessarily correct. The course of an information dialogue is not determined by what is actually true, but by what the participants *believe* to be true. What is meant here by saying that *S knows that x*, is no more than that *S has the information x available*, without implying any commitment to the truth of x. Until further notice we will use the terms 'know that' and 'believe that' interchangeably, as shorthands for 'to have the information available that'.

Natural information dialogues contain a substantial amount of verification, which indicates that participants in such dialogues often have uncertain knowledge about something. I will describe the situation where a participant S has some information p available without fully trusting it as *S suspects that p*. Not only the information available to the partners is crucial in an informative dialogue, but also the information which is *not* available and in particular the information which they want to *become* available. There are two ways in which one may want information to become available: one may want it to become available to oneself or one may want it to become available to the partner. In other words, one may want to know something or one may want to make something known. In information dialogue, these are the only possible intentions that can underly a communicative act.

On the basis of an analysis of the flow of information in actual information dialogues, in Bunt (1986) a taxonomy of communicative action types has been developed using three major categories: questioning, informing, and answering acts. In each of these categories a variety of communicative functions is distinguished, which are characterized in terms of packages of appropriateness conditions. The functions within one category share an appropriateness condition that expresses the intention motivating the act. For questioning acts this is the condition that the speaker wants to know something (namely the value of the semantic content of the utterance), for informing acts it is that the speaker wants to make something known to the addressee, and for answering it is the speaker's knowledge that the addressee wants to know something. The details of this are of no special concern here; important is that each utterance is supposed to realize at least one communicative act, which conveys a certain package of beliefs and intentions on the part of the speaker. In describing these packages, it is useful to make the well-known distinction between semantic content and communicative function, where the communicative function characterizes the way in which the belief- and intention attitudes of the dialogue partners are involved. The attitudes most relevant for information dialogues are:[1]

(2) *to know that ..*
to suspect that ..
to want to know ..
to want to make known that ..

As we have seen, these attitudes are relevant in connection not only with factual information but also with information about aspects of speaker's and addressee's states, in case of dialogue control acts where the speaker verifies an intention, checks availablity of information, etc. Therefore, these attitudes more often than not occur in combinations like *U wants to know whether S knows ...*

[1]Below we shall consider the question whether a complex attitude like *X wants to know that ..* is better split up into simpler attitudes like *X wants that X knows that ..*

In fact, according to the analysis of communicative action in Bunt (1986), for an addressee who builds a model of the speaker these attitudes *always* occur in combination, even if the speaker performs an act that provides or requests factual information. For what information does an utterance like *It's raining*, when used to inform the addressee, convey? The answer is: it depends. If the addressee can look out of the window and see that the sun is shining, he will not believe that it's raining. He will, instead, think that the speaker mistakenly believes that it's raining. The interesting point about this is that, if the addressee interprets the speaker's utterance as an inform-action, he will believe that the speaker believes it's raining and wants to make that known to the addressee - a separate decision on the part of the addressee is whether to accept the factual information forming the speaker's belief or not; that will depend on the availability of other information. Characteristic of an inform (as opposed to a question or a lie) with semantic content p is that the speaker believes that p and wants to make p known to the partner; therefore, the recognition of the speaker's action will add the following elements to the addressee's model of the speaker:

(4) *Addressee knows that Speaker knows that p*
Addressee knows that Speaker wants to make known to Addressee that p

Moreover, when subsequently the addressee indicates (explicitly, by means of a dialogue control act, or implicitly) that he has understood the speaker's act, this means for the speaker that the elements (4) have indeed been added to the addressee's model of the speaker; therefore, the speaker now adds to his model of the addressee:[2]

(5) *Speaker knows that Addressee knows that Speaker knows that p*
Speaker knows that Addressee knows that Speaker wants to make known to Addressee that p

In fact, this line of reasoning can be continued indefinitely, leading up to the thesis that successful communication is achieved only when an utterance gives rise to *mutual knowledge* concerning the speaker's intentions and beliefs (see Clark & Marshall, 1981).

[2] To be quite precise, some chronological marking would have to be added.

3 User models, data bases, and logical models

3.1 Requirements on user models

From the above observations we can derive certain requirements on the kind of models that are needed to represent the state of a partner in an information dialogue, in particular that of the human user of an intelligent interactive information system.

The first and foremost requirement on such user models is that they should be capable of representing epistemic intentions and recursive information about such intentions as well as about both certain and uncertain information ('knowledge/belief' and 'suspicion').

Second, this information is typically incomplete at all levels: there is usually incomplete information about what the user intends, knows and does not know. So the model should represent incomplete information adequately.

Third, closely related to the previous point, the model should handle incomplete information in a computationally effective way. Typically, the information about the user's knowledge and intentions is very limited at the beginning of the dialogue; as the dialogue goes on, more and more information becomes available. To handle this in a computationally attractive way, these models should not have an eliminatory character, as standard logical possible-worlds models do, but should be incremental in character.

Fourth, the models should be suited to updating. The interpretation of a given utterance typically affects only certain 'dimensions' of a user model, namely those corresponding to the attitudes with which the factual information in the utterance is associated, according to the utterance's communicative function. Therefore, they should be organized in a modular fashion with respect to the relevant propositional attitudes.

Fifth, the model should take the chronology of the user model changes to some extent into account. A user engaged in a dialogue may suspend certain intentions, discard certain information which he later may want to reconsider, etc.

In the following sections we will outline a formalism for defining models which satisfy at least the first four requirements; the chronological aspect will not be considered here.

3.2 User modelling and data bases

For realistic applications, the system's knowledge of the discourse domain will usually be contained in a data base. A well-defined data base is fruitfully viewed as a semantic model of the formal language in which the knowledge in question can be expressed (see Bronnenberg et al., 1979; Konolige, 1981). Whether user models can be treated in a similar way is not obvious.

A fairly old idea in AI is to represent the system's knowledge of the user's knowledge of the discourse domain in a separate data base. This idea runs into difficulties, however, if we take into account that such knowledge is typically incomplete.

Suppose a system S incorporates a data base D of elementary facts about the discourse domain; S 'knows' those facts plus all the complex facts that evaluate to TRUE when broken down into facts in D. Let D' be the data base containing the elementary facts S believes a certain U, who communicates with S, to know. If S does not know whether U knows that p, we omit p from D'. But now suppose S knows that U does *not* know that p; in that case it does not help to omit p somewhere.

Absence of knowledge concerning a certain proposition, disjunctive knowledge (such as *S knows that p or q*), negative knowledge (*S knows that John does not live in Paris*), and conditional knowledge pose serious problems for a multi-database representation system. Moore (1980), who discusses these problems, concludes that *"There may be ways to get around these difficulties, but it is clear that any adequate solution is going to be much more complex than 'just using data bases'."* Effectively, it seems that the multi-data base idea has no longer been pursued in recent years.

One possible solution might be to use a *set* of data bases to represent the knowledge of U, rather than a single one, and to stipulate that the facts known to U are those which are true in each of these data bases. Thus, to represent that S knows that p, that S knows that U knows that q, and that S knows that U does not know whether q, we might use the configuration of data bases shown in Fig.1, where D'1 and D'2 together represent the knowledge of U.

```
                      ---------
                      |   p   |
                      |   q   |D'1
       ---------      |       |
       |  p    |--U-->---------
--S--> |       |
       |       |--U-->---------
       ---------      | not-p |
           D          |   q   |D'2
                      |       |
                      ---------
```

Fig.1. S knows that p;
S knows that U knows that q;
S knows that U does not know whether p.

The use of multiple data bases bears some similarity to the possible-worlds approach of epistemic logic, the standard way of representing knowledge (or belief) in logic.

3.3 Possible worlds and partial models

In the possible-worlds approach to knowledge (or belief), knowledge is expressed in terms of the alternative worlds that the agent considers possible. For an agent S, those worlds are distinguished among the set of all logically possible worlds by taking part in the relation 'accessible for S'. When S knows that p, this is represented by p being true in all worlds accessible for S; when S knows that *not* p, this is represented by p being false in all S-accessible worlds. When S does not know whether p, there is at least one S-accessible world where p is true and one where it is false.

This approach is too inefficient for computer implementation, since S not knowing the truth of a fact q which has not been considered before, is modelled by adding to each S-accessible world one where q is true and one where q is false. Therefore, the less an agent knows, the more worlds have to be represented. Moreover, the facts whose truth an agent does know have to be represented as such in *every* one of these worlds. All in all, this has the effect that for a realistic domain of discourse, with a large number of potential facts, the representation of an agent's incomplete knowledge involves an astronomic number of sets of facts; moreover, each of these sets is large, since possible worlds are *complete*: every atomic proposition must have a truth value in every world.

Ideally, one would prefer to model an agent's knowledge by representing only the facts he knows, and to represent these only once. This leads to *partial models*, where truth values are assigned to only those propositions whose truth is known. Incomplete data bases, which occur in the multiple-data base approach, are natural implementations of partial models. Partial models get complicated, however, because the modelling of disjunctive, negative and conditional knowledge, as well as of knowledge about absence of knowledge, is not straightforward and gives rise to similar difficulties as the multi-database approach (see Moore, 1980 and Bunt, 1989).

We have developed an approach which constructs modular partial models using structured clusters of (partial) valuation functions, somewhat similar to Fagin, Halpern and Vardi's 'knowledge structures' (Fagin, Halpern & Vardi, 1984; Fagin & Vardi, 1985); the precise relation between these structures and our partial models is explored in Jaspars (1989). For a finite domain of discourse, each valuation function can be implemented as a miniature data base containing its extension. A structured cluster of valuation functions can be used to describe the information relating to one particular combination of agents and attitudes (such as S's knowledge about U's knowledge about S's knowledge) and implemented as a small cluster of miniature data bases; we refer to those clusters (or to the function clusters they implement) as *data modules*. The entire model can be viewed as a network of data modules; we therefore call our approach the *Data Module Net* (DMN) approach.

4 Incremental partial models

4.1 Partial valuations and incomplete information

A consequence of considering the beliefs of two different agents is that the valuation of propositional terms must be agent-dependent. Moreover, this dependency cannot take the form of an 'agent' coordinate with just two values, since we also have to deal with the beliefs of one agent about those of the other (and so on). So if $V(s,p)$ denotes the truth value of p according to S, and $V(u,p)$ that according to U, we also need something like $V_{su}(p)$ for the truth value that S believes that U believes p to have. And so on. We will return to the agent-dependency of the valuation below, but first consider some properties of the valuation $V(\alpha, ..)$ for a fixed value α of the agent-coordinate. We will use the notation V_α to indicate this subfunction of the valuation.

The ideal of only representing the facts that an agent knows has the consequence that, if agent α does not know whether the atomic proposition p is true or false, V_α should be undefined for p, i.e. V_α should be a partial function. However, just making the valuation functions partial is insufficient for dealing with incomplete information in general. If agent α knows that *p or q*, but not which of the two, this cannot be represented by the partiality of V_α alone. What we do in this case is introduce two *'alternative extensions'* of V_α, one that makes p true and one that makes q true. Calling these extensions V_{α_1} and V_{α_2}, we have, formally:

(5) $$\begin{aligned} V_{\alpha_1}(x) &= V_\alpha(x) \text{ for } x \neq p, \text{ and } V_{\alpha_1(p)} = 1 \\ V_{\alpha_2}(x) &= V_\alpha(x) \text{ for } x \neq q, \text{ and } V_{\alpha_2(q)} = 1 \\ V_\alpha &\sqsubseteq V_{\alpha_1};\ V_\alpha \sqsubseteq V_{\alpha_2} \end{aligned}$$

This implements the view that, if S knows that *p or q* but knows neither that p nor that q, upon extending his knowledge S will obtain the knowledge that p or the knowledge that q (or both).

Agent α's knowledge that *p or q* can now be modelled by V_α having these alternative extensions V_{α_1} and V_{α_2}, and the truth condition (6):

(6) α *knows that* $p \vee q \iff$ $V_\alpha(p) = 1$ or $V_\alpha(q) = 1$ or in all alternative extensions V_{α_i} of V_α : $V_{\alpha_i}(p) = 1$ or $V_{\alpha_i}(q) = 1$

(This will be made more precise below.) Figure 2 gives a pictorial representation of a model which represents, according to (6), that S *knows* (only) *that p or q*.

```
                          ---------
                          |       |
                          |   p   |
          ---------       |       |
          |       |--1-->---------
   --S--> |       |
          |       |--2-->---------
          ---------       |       |
                          |   q   |
                          |       |
                          ---------
```

Figure 2. S knows that p or q

We see here the beginning of the use of 'function clusters' (or 'data modules'): it is the valuation V_α *with its alternative extensions* V_{α_i} that represents α's information about the domain of discourse.

Note that, by the stipulations $V_\alpha \sqsubseteq V_{\alpha_1}$; $V_\alpha \sqsubseteq V_{\alpha_2}$, the valuations V_{α_1} and V_{α_2} indeed constitute *extensions* of the knowledge represented by V_α. In other words, the knowledge of V_α is ***persistent in alternative extensions.*** We do not want to copy the extension of V_α in its alternative extensions, though; that would go against our aim to represent the facts an agent knows only once. We therefore define alternative extensions slightly differently by (7) as valuating only the *additional* facts, and we will define the truth conditions of 2a-DpL expressions in such a way that they ensure the persistence of information in alternative extensions.

(7) $V_{\alpha_1}(x)$ is undefined if $V_\alpha(x)$ is defined, and $V_{\alpha_1(p)} = 1$
$V_{\alpha_2}(x)$ is undefined if $V_\alpha(x)$ is defined, and $V_{\alpha_2(q)} = 1$

'Negative information' gives rise to similar complications. Suppose agent α has very little information about the nationalities of the persons in a certain domain of discourse, not knowing anybody who's Italian, for instance, but knowing that John is not Italian. So α's information about the predicate constant ITALIAN is restricted to that the individual john does not belong to $V_\alpha(ITALIAN)$. This 'negative information' can be modelled by introducing, in addition to V_α, a 'negative part' $V_{\alpha\, neg}$ that expresses which entities do *not* belong to the extensions of the terms.

As long as the 'embedded' language expressing the objects of doxastic and intentional operators is propositional logic, the need to represent negative information does not present itself at the level of terms.[3] However, we need to be able to express negative *complex* information and intentions. This brings us back to the agent-dependency of the valuation.

Using the subfunction V_{su} for S's beliefs about U's beliefs, $V_{su}(p) = 1$ represents that *S believes that U believes that p*, and $V_{su}(p) = 0$ that *S believes that U believes that not p*; we also need a way to represent that *S believes that U does not believe that p*. To this end we introduce the additional valuation $V_{s\, neg\, u}$, which expresses S's negative information about U's information.

Alternative extensions, as introduced above, can take care of disjunctive

[3]For instance, knowing that a propositional constant p is not true is equivalent to knowing that it is false.

factual knowledge. The agent-dependency of the valuation must also take alternative extensions into account, because an agent may have disjunctive information involving the other agent. For instance, if all that S knows is that *if U knows that p then q*, we need the alternative extensions V_{s_1} and V_{s_2} such that $V_{s_1}(q) = 1$ and $V_{s_2\,neg\,u}(p) = 1$. Figure 3 gives a pictorial representation of this model.

```
                      ---------
                      |       |
                      |   q   |
      ---------       |       |
      |        |--1-->---------
--S-->|        |
      |        |--2-->---------
      ---------       |       |
                      |       |
                      |       |
                      ---------
                          |
                         neg
                          |
                      ---------      ---------
                      |       |      |       |
                      |       |--U-->|   p   |
                      |       |      |       |
                      ---------      ---------
```

Figure 3. S knows that if U knows that p then q

5 DMN models

In this section we first describe DMN models for the simple case where two agents are taken into account and one propositional attitude, viz. *to have the information available that* (for the sake of brevity also referred to as *to know* or *to believe*). and where the embedded logic is propositional logic. So we do not consider the internal structure of factual information here. The corresponding logical language we call 2a-DpL: Two-agent Doxastic propositional Language. In the next section we will add intention operators.

5.1 Formal definition of DMNs

A DMN model can be viewed as a network of modules (function clusters) linked together through inter-speaker connections. If the model is to represent the information state of agent α, it contains a module with index α which functions as an 'entry' to the network (like S in Fig. 3). The presence of alternative extensions depends on which disjunctive knowledge is to be represented; the specification of these extensions will therefore be part of the model. This leads to the definition of a DMN model (or 'DMN') as a tuple consisting of 3 elements: (1) a set of agent-dependent partial valuations; (2) the particular 'entry point' valuation describing the domain knowledge of the agent whose state of information is modelled; (3) the specification of alternative extensions.

DEFINITION. **A DMN model for 2a-DpL** is a triple

(8) $M = < F_\alpha, \mathcal{F}, \mathcal{A} >$, where:
- $F_\alpha \in \mathcal{F}$;
- $\mathcal{F}$ is an indexed set of partial functions from the propositional letters of 2a-DpL to truth values;
- $\mathcal{A}$ is a partial function from $\mathcal{F}$ into $\mathcal{P}(\mathcal{F})$ (specifying the alternative extensions present in M)

In what follows we will mostly use 'S' as a name of the agent whose information states are modelled, and U as that of the other dialogue participant. The indices occurring in the indexed set of functions $\mathcal{F}$ are defined as follows.

DEFINITION. For a model $M = < F_s, \mathcal{F}, \mathcal{A} >$ **the set I_M of indices** is the smallest set such that:

(9)
1. s belongs to I_M;
2. if $i \in I_M$ then is and iu belong to I_M;
3. if $i \in I_M$ and i is not of the form $j\,neg$, then $i\,neg \in I_M$;
4. if $i \in I_M$ and $F_k \in \mathcal{A}(F_i)$ then $i_k \in I_M$.

Stipulating that a certain index i belongs to I_M is in fact a way of stipulating that the function F_i belongs to $\mathcal{F}$. Since the definition of the set of indices of a model is the same for every 2a-DpL model, we keep it for the sake of readability outside the model definition (8).

Note that the specification of the inter-speaker connections and negative parts does not form part of the model. This is because we assume that every valuation has both a positive and a negative part and has connections to both agents; however, many of these connections will connect up to 'empty' parts, which contain no information.

We can now make precise what we mean by a 'function cluster' (or 'data module').

DEFINITION. A **function cluster** or **data module** in a DMN-model M_i (for 2a-DpL) is the information contained in a valuation F_i, for some index $i \in I_M$, plus that in the negative part F_{ineg} and that in the alternative extensions F_{i_k}, for $i_k \in I_M$.

The index i of the valuation F_i which forms the heart of a function cluster will be called the index of the function cluster.

A *DMN submodel* is a (sub-)DMN with an 'entry point' corresponding to a particular complex agent/attitude combination α, such as *S believes that U believes that S does not know that* (where the 'entry' index would be *s u neg s*). We will write M_i to denote the sub-DMN with entry index i.

Formally, a sub-DMN of a DMN-model $M = < F_\alpha, \mathcal{F}, \mathcal{A} >$ is a triple $< F_i, \mathcal{F}', \mathcal{A}' >$ where $F_i \in \mathcal{F}'$, $\mathcal{F}' \subseteq \mathcal{F}$, and $\mathcal{A}'$ is $\mathcal{A}$ restricted to $\mathcal{F}'$.

We now turn to the semantics of 2a-DpL expressions of the form $S \ \Vdash X$ ('*S has the information that X*'), i.e., we define under what conditions a DMN represents that S knows that X.

5.2 Modularity and locality in DMN models

One of the requirements on models that we formulated above is that of *modularity*. In order to determine whether *S knows that U knows that p*, in the possible-worlds approach one has to consult all the worlds which are U-accessible from a world that is S-accessible. S's knowledge about U's knowledge is, so to say, spread out over the entire model, intertwined with information relating to other propositional attitudes. The computational complexity of evaluation and update operations may greatly benefit from a more modular organization, where one consults only the relevant 'modules' of the model. DMN-models have been designed to meet this requirement: to determine, for example, whether (10) is true in a DMN-model one consults the function cluster with index *su* (the module containing what S knows that U knows).

(10) *S knows that U knows that q*

In very simple cases, evaluation and update indeed involve only one function cluster. In general, however, the consultation of several 'modules' is required. This is in fact even the case for (10), as the model of Fig. 4 illustrates.

```
                                      ---------
                                      |       |
                                      |   p   |
         ---------      ---------     |       |
         |       |      |       |--1-->---------
--S-->|  |       |--U-->|       |
         |       |      |       |--2-->---------
         ---------      ---------     |       |
                            |         |   q   |
                           neg        |       |
                            |         ---------
                        ---------
                        |       |
                        |   p   |
                        |       |
                        ---------
```

Figure 4. S knows that U knows that (p or q);
S knows that U knows that not p

This model represents the information (11):

(11) *S knows that U knows that (p or q)*
S knows that U knows that not p

Since (10) can be deduced from (11), we want the evaluation of S knows that U knows that q to come out true.

The model says that, according to S, U entertains two possibilities, corresponding to the submodels M_{su_1} and M_{su_2}. However, in both cases U holds the belief that *not p*, in view of the persistence of information in alternative extensions. So the two possibilities that U entertains correspond in fact to the *totalities* of information available at the index sets $\{su,\ su_1\}$ and $\{su,\ su_2\}$.

When we consider the first of these sets, we are dealing with a situation where, according to S, on the one hand U knows that *not p* since $F_{su\,neg}(p) = 1$, but on the other hand U knows that p since $F_{su_1}(p) = 1$. Therefore, this submodel can be viewed as an *inconsistent part* of the DMN. The only possibility that S can entertain consistently corresponds to the index set $\{su, su_2\}$. We therefore define a DMN model to represent a belief on the part of S only if that belief is true according to the relevant *consistent* subsets of valuations.

It turns out that the collections of valuations which are relevant in the truth-conditional semantics of 2a-DpL are of a the same kind as the collections that we see in this example. Such collections of function clusters form small parts of DMN-submodels, which we call *truncated submodels*. Their precise definition is as follows.

DEFINITION. A **truncated submodel** of a submodel $M_\alpha = < F_\alpha, \mathcal{F}, \mathcal{A} >$ is a triple $T = < F_i, \mathcal{F}', \mathcal{A}' >$, where $\mathcal{A}'$ is $\mathcal{A}$ restricted to $\mathcal{F}'$ and $\mathcal{F}'$ is the smallest subset of $\mathcal{F}$ defined by:

(i) $F_i \in \mathcal{F}'$;

(ii) if $F_j \in \mathcal{F}'$ then $F_{j\,neg} \in \mathcal{F}'$ and $F_m \in \mathcal{F}'$ for every $F_m \in \mathcal{A}'(F_{j\,neg})$;

(iii) if $\mathcal{A}(F_i) \neq \emptyset$ then there is exactly one $F_k \in \mathcal{A}(F_i)$ such that $F_k \in \mathcal{F}'$;

(iv) if $F_j \in \mathcal{F}'$, where j is a complex index decomposable as $j = g\,h$, with $h = s$ or $h = u$, and if $\mathcal{A}(F_g) \neq \emptyset$, then there is exactly one $F_k \in \mathcal{A}(F_g)$ such that $F_{kh} \in \mathcal{F}'$.

The index i of the truncated submodel $< F_i, \mathcal{F}', \mathcal{A}' >$ will be called *the index of the truncated submodel.*

In words, a truncated submodel with index i is that part of the submodel that contains the information locally available at index i, plus what is available at one of the alternative extensions at that index; moreover, if the index is complex, say $i = sus$, we also take the last step back in building up i, take one alternative extension at the shorter index, say su_3, and add to that the operation used to build i, so in the example we get su_3s.

The definition of when a 2a-DpL expression of the form S $\Vdash \phi$ is true in a DMN model will be given recursively in terms of the truth in certain consistent truncated submodels. The set of all consistent truncated submodels with index i of a given (sub-)model M_α will be denoted by $CTS(i)$.

The fact that the semantics of 2a-DpL can be formulated in terms of truncated submodels emphasizes that DMN-models to a great extent work in a modular and local fashion.

5.3 Truth in a DMN model

A DMN model M as a whole behaves classically in that any formula of the form S $\Vdash$ ϕ is either true or false in the model. Truncated submodels, on the other hand, have a three-valued logic: given a consistent truncated submodel T, a 2a-DpL formula ϕ can be true in T, false in T, or undefined in T. We will use the notation $T \models\!\!\equiv \phi$ in the first case and $T \equiv\!\!| \phi$ in the second.

The definition starts off at the level of M as follows:

DEFINITION. The **truth of a 2a-DpL formula expressing a belief on the part of S in the DMN-model** M is defined by:

$$M \models S \Vdash \phi \iff T \models\!\!\equiv \phi \text{ for every } T \text{ in } CTS(s), \text{ and } CTS(s) \neq \emptyset.$$

In defining the truth conditions relative to truncated submodels the notion 'consistent U-extensions' of a truncated submodel is used. A U-extension of a truncated submodel with index i is a truncated submodel with index iu; we use the notation $CUX(T)$ to denote the *consistent* U-extensions of T; similarly for S-extensions.

DEFINITION. The **truth of a 2a-DpL formula in a truncated submodel** T_i is defined as follows.

(12) For any truncated submodel $T = < F_i, \mathcal{F}, \mathcal{A} >$:

1. If p is a propositional constant:

$$\begin{array}{lll} T \models\!\!\equiv p & \iff & F_i(p) = 1 \text{ for some } F_i \in \mathcal{F}, \text{ and} \\ & & \text{for all } F_i \in \mathcal{F}_T\text{: } F_i(p) \neq 0 \\ T \equiv\!\!| \; p & \iff & F_i(p) = 0 \text{ for some } F_i \in \mathcal{F}, \text{ and} \\ & & \text{for all } F_i \in \mathcal{F}_T\text{: } F_i(p) \neq 1 \end{array}$$

The remaining clauses apply to any (consistent) truncated submodel T and 2a-DpL expressions ϕ, ψ.

2. $T \models \phi \,\&\, \psi \iff T \models \phi$ AND $T \models \psi$
$T \,\rightmodels\, \phi \,\&\, \psi \iff T \,\rightmodels\, \phi$ OR $T \,\rightmodels\, \psi$

3. $T \models \phi \vee \psi \iff T \models \phi$ OR $T \models \psi$
$T \,\rightmodels\, \phi \vee \psi \iff T \,\rightmodels\, \phi$ AND $T \,\rightmodels\, \psi$

4. $T \models \neg\phi \iff T \,\rightmodels\, \phi$
$T \,\rightmodels\, \neg\phi \iff T \models \phi$

5. $T \models U \Vdash \phi \iff$ for all consistent U-extensions T' of T: $T' \models \phi$, and $CUX(T) \neq \emptyset$
$T \,\rightmodels\, U \Vdash \phi \iff$ for all consistent U-extensions T' of T: $T' \,\rightmodels\, \phi$ and $CUX(T) \neq \emptyset$
Similarly for $T \models S \Vdash \phi$ and $T \,\rightmodels\, S \Vdash \phi$.

6 Adding intentions

6.1 Intentional attitudes

The 'boulomaic' propositional attitudes of intention, desire, goal, want, etc. have been studied much less than the epistemic-doxastic attitudes of knowledge and belief, and their logic is correspondingly underdeveloped. Boulomaic attitudes come in subtly varying forms and present tricky logical problems. When we restrict ourselves to information dialogues, however, we can focus on particular intentional attitudes of which a formalization appears to be feasible.

An important general difference between intentions and goals on the one hand, and wants and desires on the other, is that the latter can conflict with what one believes to be possible, whereas the former cannot. One can very well *desire* to become the king of France, even though one does not believe this to be possible; to *intend* or to *have the goal* to become king of France is not well possible, however. Also, one can have different, conflicting desires, but it does not seem possible to have conflicting intentions. Another difference is that intentions and goals can only relate to situations over which one has some control. One can desire that the sun shines, but one cannot intend it.

Intentions can be 'static', such as the intention to stay home tonight; or 'dynamic', such as the intention to write a paper. Intentions of the latter kind may be called 'goals'; inherent to the nature of a goal is that it has not (yet) been achieved: as soon as a goal has been achieved, it is no longer a goal. Dynamic intentions (goals) give rise to actions, and are of special interest here because communicative actions have such intentions, and their interpretation crucially involves the understanding of what these intentions are. Indeed, we assume that in an information dialogue every communicative action owes its existence to some underlying intention, and that the logic of these intentions is largely responsible for the 'logic' of these dialogues.

What sort of intentions can one have in an information dialogue? Since the participants in such a dialogue by definition are supposed to have no other purposes than exchanging factual information, we can *à priori* identify two kinds of possible goals for a participant: the situation where participant has obtained certain factual information, or the situation where the partner has obtained certain factual information. We have also seen that participating in a dialogue gives rise to situations where one wants to clarify, verify, explain, etc., i.e. where one wants to obtain or to provide information which is not of a factual nature (not relating to the domain of discourse), but of a 'communicative' nature, relating more to the communication as such. So, the kinds of goals that arise for an agent in an information dialogue are:

A the agent possesses certain information, factual or communicative, which he did not posses before;

B the dialogue partner possesses certain information, factual or communicative, which he did not posses before.

When we try to formalize such intentions we may want to take into account that expressions like S *intends that* ϕ are well-formed only if ϕ expresses one of the two types of intention just mentioned. So an expression like S *intends that* $\neg ITALIAN(john)$ is incorrect. Since an intention operator may only occur in combination with obtaining or providing information, one possibility is to introduce attitudes for these combinations: the attitudes *wanting to know something* and *wanting to make something known*. This possibility has been explored in Bunt (1990), where the first of these attitudes is construed mathematically as an *extension* of what the agent in question knows (since one can only want to know something one doesn't know); the second as a *restriction* of what one knows. The latter decision means that we assume one can only make something known which one knows, which is reasonable if we restrict ourselves to information dialogues.

Using the complex attitudes *wanting to know something* and *wanting to make something known*, as opposed to the simpler attitudes *to want* and *to* *know*, has the drawback that it does not allow the correct formulation of the desired logical properties. A property one would for instance like a *want* attitude to have is that, if an agent S wants that p and knows that p implies q, then he does not want *not* q; using $\mathrel{|\!\approx}$ to denote ***wants that*** and $\Vdash$ to denote ***know that***:

$$(13)\quad S \mathrel{|\!\approx} p \quad \& \quad S \Vdash (p \to q) \quad \Longrightarrow \quad \neg S \mathrel{|\!\approx} \neg q$$

Use of the two complex attitudes ***wanting to know something*** (denoted $\mathrel{|\!\!\leftarrow}$) and ***wanting to make something known*** ($\mathrel{\|\!\!\mapsto}$), would lead to two instances of (13), viz.:

$$(14)\quad \begin{array}{lllll} S \mathrel{\|\!\!\mapsto} p & \& & S \Vdash (p \to q) & \Longrightarrow & \neg S \mathrel{\|\!\!\mapsto} \neg q \\ S \mathrel{|\!\!\leftarrow} p & \& & S \Vdash (p \to q) & \Longrightarrow & \neg S \mathrel{|\!\!\leftarrow} \neg q \end{array}$$

The latter of these instances is incorrect, however, since wanting to know whether $\neg q$ is logically equivalent to wanting to know whether q; and is certainly wrong to require that an agent does *not* want to know any (known) implications of what he wants to know. Therefore, in Bunt (1990) the alternative property (15) is suggested, which however is not quite satisfactorily either, as it is too strong to require that an agent wants to know *all* the (known) implications of what he wants to know.

$$(15)\quad S \mathrel{|\!\!\leftarrow} p \quad \& \quad S \Vdash (p \to q) \quad \Longrightarrow \quad S \mathrel{|\!\!\leftarrow} q$$

The problem is that the complex attitude *wanting to know*, treated as primitive, does not allow the use of a negation inside the scope of the *want* part but outside the scope of the *know* part. This is one of the reasons why in the present paper we will introduce a separate attitude *intend.* (Another reason is that it becomes possible to express that an agent wants the partner to believe that p without it being logically necessary that the agent himself believes that p.)

Beun (1989) gives the following axioms for an intention operator I_s (S *intends that*), where B_s stands for S *believes that*:

$$(16)\quad \begin{array}{lll} I_s p & \Longrightarrow & \neg B_s p \\ I_s p \;\&\; B_s(p \to q) & \Longrightarrow & \neg I_s \neg q \\ I_s p & \Longrightarrow & B_s I_s p \\ \neg I_s p & \Longrightarrow & B_s \neg I_s p \end{array}$$

These axioms are particularly attractive if we interpret the I_s operator as expressing a dynamic intention (or 'goal') on the part of S, and B_s as our attitude *to have available the information that.* The first axiom expresses that an agent's goal is not believed by him to be achieved already; the second that one's intentions are consistent with one's information (and, as a consequence, that different intentions do not conflict); the remaining two axioms express full introspective knowledge about one's intentions.

We will take the intention attitude underlying communicative acts in information dialogues to have these properties. It immediately follows, for example, that the intention to know whether p is equivalent to the intention to know whether $\neg p$:

$$(17) \quad S \mathrel{|\!\!\approx} S \vdash p \quad \Longrightarrow \quad S \mathrel{|\!\!\approx} S \vdash \neg p$$

Here we have again used $|\!\!\approx$, now to denote more specifically the dynamic intention attitude we want to have, and $\vdash$ to denote 'knowing whether' (formally just an abbreviation of 'knowing that or knowing that not').

We will call the language 2a-DpL, extended with this intentional operator: 2a-DIpL (*Two-agent Doxastic-Intentional propositional Language*). We now turn to the representation of intentions in DMN models, which is the same as specifying the semantics of the intention attitude.

6.2 Intentions in DMN models

Let us begin with an informal sketch of a user model in DMN style where a system S has available the information about user U that U knows that p and r, and U has the intention that the system knows that r; moreover, S has available the factual information that p, q, and t, and S intends U to know that q.

```
                          ---------
                          |   p   |
                          |   r   |
        ---------         |       |
        |   p   |---u--->---------
---s--->|   q   |
|       |   t   |-- u!-->---------        ---------
|       ---------         |       |        |       |
|                         |       |---s--->|   r   |
|                         |       |        |       |
|                         ---------        ---------
|       ---------         ---------
|       |       |         |       |
|--s!-->|       |---u--->|   q   |
        |       |         |       |
        ---------         ---------
```

Fig. 5. S knows that p, q, and t;
S knows that U knows that p and r;
S knows that U intends that S knows that r.

Here we have used the labels $s!$ and $u!$ to mark the connections to data modules that express intended states of information. The picture immediately reflects that this model represents not only beliefs but also intentions, and suggests that such a DMN has two entry points (marked s and $s!$) instead of one; therefore, one would expect such a DMN to be defined formally as a quadruple rather than a triple, as in definition (8). This is not so, however, since the valuation function corresponding to the entry index $s!$ is always empty, in view of the fact that the intentions in information dialogues always concern the knowledge of one of the participants, rather than factual information. The only thing that has to change in the DMN definition is that of the index set characterizing the set of partial valuations). Instead of the first two clauses of definition (9) we now get:

(18) 1. s and $s!$ belong to I_M;
2. if $i \in I_M$ then is, $is!$, iu and $iu!$ belong to I_M;

To the truth definitions given in the previous section we have to add the definitions of the truth in a model and in a truncated submodel of expressions of the form $X \approx\!\!\!\!| \;\phi$. This is rather straightforward. The definition of truth in a model is entirely analogous to the one given in the previous section:

$$M \models S \mathrel{|\!\!\approx} \phi \iff T \models \phi \text{ for every } T \text{ in } CTS(s!), \text{ and } CTS(s!) \neq \emptyset.$$

To define truth in a truncated submodel, we first note that, with the amendment (18), the definitions of submodel and truncated submodel remain unchanged. Next we define 'S!'- and 'U!'- extensions of truncated submodels by analogy with 'S-extension' and 'U-extension', and stipulate:

(19) For any truncated submodel $T = < F_i, \mathcal{F}, \mathcal{A} >$:

$$
\begin{array}{lll}
6. \quad T \models U \approx \phi & \Longleftrightarrow & \text{for all consistent } U!\text{-extensions } T' \text{ of } T\text{:} \\
 & & T' \models \phi, \text{ and } CU!X(T) \neq \emptyset \\
T \dashv U \Vdash \phi & \Longleftrightarrow & \text{for all consistent } U!\text{-extensions } T' \text{ of } T\text{:} \\
 & & T' \dashv \phi \text{ and } CUX(T) \neq \emptyset, \text{ or} \\
 & & \text{for all consistent } U!\text{-extensions } T' \text{ of } T\text{:} \\
 & & T' \models \phi, \text{ and } CU!X(T) \neq \emptyset
\end{array}
$$

Note that the last clause in this rule accounts for the fact that one cannot have a goal that one believes to be satisfied, as expressed in the first axiom of (16). Similar clauses can be added to take the other axioms into account.

7 DMN models and user modelling

7.1 Limitations of 2a-DIpL models

The DMN models defined above are too simple to be immediately useful for user modelling.

First, 2a-DIpL models are designed for the use of the propositional-logical language for describing factual information, and therefore disregard the internal structure of such information. For practical use we need DMN models for a more powerful language, with predicate-argument structures, quantifiers, modal operators, non-individuating expressions (mass terms, collectives), etc.

Second, we have mentioned earlier that besides the kind of belief characterized as *to have the information that* we must also take weak belief (*'suspicion'*) into account. The 'suspicion' attitude can be handled in much the same way as the attitude *to have the information that*, which from now on we will call *to know*. (We will use *beliefs* generically to refer to knowledge and suspicions.) Since our interpretation of *know* is nonfactual, it has much the same logical properties as *suspect*. There are only minor differences, such as the introspection axioms: for *know* this is (20a); for *suspect* this is not the corresponding formula (20b), but (20c):

(20a)	*S knows that p*	$\Longrightarrow$	*S knows that S knows that p*
(20b)	*S suspects that p*	$\Longrightarrow$	*S suspects that S suspects that p*
(20c)	*S suspects that p*	$\Longrightarrow$	*S knows that S suspects that p*

Properties like (20a have not been built into the truth definitions given in the previous section, but they might be.[4]

In addition, the reltaions betweeen the ***suspect*** attitude and the other attitudes deserves some consideration. For instance, an agent's suspicions should not contradict his knowledge: if *S **knows that** p*, then it should not be the case that *S **suspects that not** p*. This should also be accounted for in the truth definitions. We will not pursue the addition of the ***suspect*** attitude here, as it mainly has the effect of adding complexity to DMN-models and truth conditions.

Third, we have suggested in the analysis of communicative action that succesful communication leads to *mutual knowledge*. In user modelling, mutual knowledge may occur within the scope of the ***know***, ***suspect*** and ***intend*** attitudes. If *S **knows that it is mutually known by** U **and him that** ϕ*, then by definition also *S **knows that** U **knows that** ϕ*. This illustrates that the facts S knows to be mutually known by U and him form a subset of the facts S knows that U knows, as well as of the facts S knows that U knows that S knows, etc. We can thus add mutual knowledge to 2a-DIpL and its models by adding the appropriate valuations, provided that we make sure to express in the truth definitions that mutual knowledge implies nested ordinary knowledge at any depth of nesting.

7.2 DMN models and dialogue management

The importance of sophisticated user models lies in the fact that they should be the basis for a system to act 'intelligently' in compliance with an understanding of what the user wants. In particular, adequate user models are a prerequisite for intelligent *inter*action, which is the angle from which we are concerned with user models in this paper. This raises two questions here: how can the kind of models we have described be used for *generating* communicative actions, and how can they be constructed and maintained on the basis of *interpreting* incoming communicative actions? These questions cannot be answered in any detail within the scope of the present paper; in particular, the planning and generation of communicative acts on the basis

[4]In fact, they ***should*** be. There are several choices one can make for these properties, just like there are many axiomatizations of epistemic and doxastic logic.

of a user model is an issue in itself. We will however briefly indicate how a DMN model can be constructed and maintained by describing a *model update function* that defines how a given DMN should be updated as the result of interpreting an incoming communicative act. For a more detailed discussion the reader is referred to Bunt(1990).

Starting point for the definition of a model update function is the observation that, according to the analysis in section 2, a communicative act always conveys a package of information about the beliefs (knowledge and suspicions) and intentions of the speaker. Exactly which package is conveyed depends on the communicative function and the semantic content of the act. The system's understanding of the communicative function and semantic content of an incoming act should thus result in adding the corresponding information about the user's beliefs and intentions to the model. Let us assume that an interpretation module has expressed these beliefs and intentions in 2a-DIpL formulae, where 2a-DIpL is suitably extended to deal with semantic contents of greater complexity than what we can express in propositional logic. A complication is now that the model may already verify or falsify one of these formulae. If it already verifies a formula, nothing needs to be done, but what if the model falsifies the formula? It depends on 'tactical' considerations what to do in such a case; this is not so much a matter of logic or semantics, but it depends strongly on 'social' and application-specific conditions. We therefore leave this open here, defining the update function only in case the model does not contradict the formula. This covers, besides the trivial case where the model already verifies the formula, only the case where the model is underdetermined with respect to the formula. In the practice of information dialogues, this is the most important case anyway.

The update function u takes two arguments, a (sub-)model $M_i = < F_i, \mathcal{F}, \mathcal{A} >$ and a formula ϕ, and delivers a new (sub-)model $M_i' = < F_i', \mathcal{F}', \mathcal{A}' >$ that satisfies ϕ. We will use the notation $M_i[\ldots]$ to denote the submodel that is equal to M_i except (at most) for what is stipulated inside the square brackets. Note that the formulae ϕ always express knowledge of the system about the user, so they are of the form $S \Vdash \psi$).

DEFINITION. The update of a model $M_i = < F_i, \mathcal{F}, \mathcal{A} >$ with a 2a-DpL formula ϕ is defined as follows.

(21) 0. $u(M,\ \phi) \quad = \quad u(M_s, \phi)$

For any propositional constant p and index i:

1. $u(M_i,\ p) \quad = \quad M_i[F_i' = F_i \cup \{< p, 1 >]\}$

The remaining clauses apply to any DMN-submodel M_i and 2a-DpL-expressions ϕ, ψ.

$$
\begin{array}{lll}
2.\ u(M_i,\ \phi \,\&\, \psi) & = & u(u(M_i,\ \phi),\ \psi) \\
3.\ u(M_i,\ \phi \vee \psi) & = & \text{if } M_i \text{ has alternative extensions for index } i, \\
 & & \text{then for each of these, with index } j, \text{ do} \\
 & & u(M_j,\ \phi \vee \psi); \\
 & & \text{else } M_i' = M_i[\mathcal{A}(F_i) = \{F_{i_1}, F_{i_2}\}], \text{ and do} \\
 & & u(u(M_{i_1}', \phi)_{i_2}, \psi) \\
4.\ u(M_i,\ \neg\, \phi) & = & u'(M_i,\ \phi) \text{ (See below for the function } u'.) \\
5.\ u(M_i,\ U \Vdash \phi) & = & u(M_{iu},\ \phi) \\
\ u(M_i,\ S \Vdash \phi) & = & u(M_{is},\ \phi) \\
6.\ u(M_i,\ U \approx\!\!\mid\ \phi) & = & u(M_{iu!},\ \phi) \\
\ u(M_i,\ S \approx\!\!\mid\ \phi) & = & u(M_{is!},\ \phi)
\end{array}
$$

The function u' builds up the 'negative parts' of a DMN model, and is defined as follows.

$$
\begin{array}{lll}
(22)\ 1.\ \text{For propositional constant } p\text{:} & & \\
\ u'(M_i,\ p) & = & u(M_{i\,neg},\ p) \\
2.\ u'(M_i,\ \phi \,\&\, \psi) & = & \text{if } M_i \text{ has alternative extensions for index } i, \\
 & & \text{then for each of these, with index } j, \\
 & & \text{do } u(M_j,\ \neg\phi \vee \neg\psi); \\
 & & \text{else } M_i' = M_i[\mathcal{A}(F_i) = \{F_{i_1}, F_{i_2}\}], \text{ and do} \\
 & & u(u(M_{i_1}', \neg\phi)_{i_2}, \neg\psi) \\
3.\ u'(M_i,\ \phi \vee \psi) & = & u'(u'(M_i,\ \phi),\ \psi) \\
4.\ u'(M_i,\ \neg\, \phi) & = & u(M_i,\ \phi) \\
5.\ u'(M_i,\ U \Vdash \phi) & = & u(M_{i\,neg\,u},\ \phi) \\
\ u'(M_i,\ S \Vdash \phi) & = & u(M_{i\,neg\,s},\ \phi) \\
6.\ u'(M_i,\ U \approx\!\!\mid\ \phi) & = & u(M_{i\,neg\,u!},\ \phi) \\
\ u'(M_i,\ S \approx\!\!\mid\ \phi) & = & u(M_{i\,neg\,s!},\ \phi)
\end{array}
$$

In the TENDUM dialogue system (Bunt et al., 1985) a predecessor of the DMN formalism has been implemented for user modelling and dialogue maintenance, including a rule-driven strategy for the systematic generation of communicative acts. Although primitive compared to what can be done on the basis of a full DMN implementation, this may serve to illustrate how such models can be used for the generation side of communicative acts.

The TENDUM generation strategy uses the repertory of communicative acts mentioned in section 2, dividing these acts into three categories: *questioning, informing,* and *answering.* In each category there is a 'least specific' type of action, called QUESTION, INFORM and ANSWER, respectively, and a variety of more specific action types. An action type is more specific if it has additional conditions for its appropriate use. The actions in the questioning category are motivated by an intention on the speaker of the form $S \models\!\approx S \mid\!- \phi$, those in the informing category by an intention of the form $S \models\!\approx U \mid\mid\!- \phi$, and those in the answering category by $S \mid\mid\!- U \models\!\approx U \vdash \phi$. Therefore, the generation strategy begins by checking whether the current user model contains elements of one of these forms.[5] Suppose, for instance the user model M contains the element $S \mid\mid\!- U \models\!\approx U \vdash p$. In that case a rule is applied which activates a procedure for evaluating the additional conditions for generating an answering action, in particular for evaluating the condition $S \mid\!- p$, i.e., for checking whether the system knows the answer. This simply comes down to using the truth conditions described in section 5 for determining whether $M \models S \vdash p$. If this comes out true, say because the system has the information that not p, then the generator looks for the possibility to generate a more specific response (such as CONFIRM or DISCONFIRM), by trying $M \models S \mid\mid\!- U \mid\!\cdots p$ (where $U \mid\!\cdots p$ stands for: U has a suspicion about p. If this comes out false, then the system generates the action <ANSWER, $\neg p$ >. For more about the systematic generation of communicative actions, including actions with an articulate semantic content rather than 'p', see Bunt (1988).

[5] In fact, the algorithm looks first in a part of the user model which is temporarily kept separate, containing the elements added most recently as the result of processing the last incoming communicative act.

References

BEUN, R.J. (1989) *The Recognition of Declarative Questions in Information Dialogues.* Dissertation, Tilburg University 1989. Published as *ITK Research Report 14*,Tilburg: Institute for Language Technology and Artificial Intelligence ITK.

BRATMAN, M. (1989) What is intention? In: P.R. Cohen, J. Morgan & M.E. Pollack (eds.) *Intentions and Communication.* Cambridge, MA: MIT Press.

BRONNENBERG, W.J. ET AL. (1979)The PHLIQA1 question answering system. In: L. Bolc (ed.) Natural communication with computers. M'unchen: Hauser.

BUNT, H.C. (1985) *Mass terms and model-theoretic semantics.* Cambridge (UK): Cambridge University Press.

BUNT, H.C. (1986) Information dialogues as communicative action in relation to partner modelling and information processing. In: M.M. Taylor, F. Néel and D.G. Bouwhuis (eds.) (1989) *The structure of multimodal dialogue.* Amsterdam: North-Holland.

BUNT, H.C. (1987)Utterance generation from semantic representations augmented with pragmatic information. In: G. Kempen (ed.) *Natural Language Generation.* Dordrecht: Kluwer/Nijhoff.

BUNT, H.C. (1989A) Towards a dynamic interpretation theory of utterances in dialogue. In: H. Bouma & B.A.G. Elsendoorn (eds.) *Working Models of Human Perception.* London: Academic Press.

BUNT, H.C. (1989B) A model-theoretic approach to multi-database knowledge representation. *ITK Research Report 5.* Tilburg: Institute for Language Technology and Artificial Intelligence ITK.

BUNT, H.C. (1990)DIT - Dynamic Interpretation in Text and Dialogue. In: L. Kálmán (ed.) *Proceedings 2nd Symposium on Language and Logic*, Hajduszóboszló, September 1989.

BUNT, H.C. ET AL. (1985)The /sc tendum dialogue system and its theoretical basis. *IPO Annual Progress Report 19*, 105-113.

CLARK, H.H. & MASHALL, C.R. (1981) Definite reference and mutual knowledge. In: A.K. Joshi, I. Sag & B.L.Webber (eds.) *Elements of natural language understanding.* Cambridge (UK): Cambridge University Press.

FAGIN, R., HALPERN, J.Y. AND VARDI, M.Y. (1984) A model-theoretic analysis of knowledge; preliminary report. In: *Proceedings of the 25th*

IEEE Symposium Foundations of Computer Science.

FAGIN, R. AND VARDI, M.Y. (1985) An internal semantics for modal logic. In: *Proceedings of the 17th ACM Symposium on Theory of Computing*, Providence.

FAGIN, R.,HALPERN, J.Y. & VARDI, M.Y. (1984) A model-theoretic analysis of knowledge; preliminary report. In Proc. 25th IEEE Symp. Foundations of Computer Science.

FAGIN, R. & VARDI, M.Y. (1985) An internal semantics for modal logic. In Proc. 17th ACM Symp. on Theory of Computing, Providence.

HALPERN, J.Y. AND MOSES, Y. (1984) Knowledge and common knowledge in a distributed environment. In: *Proceedings of the 3rd ACM Symposium Principles of Computing.*

JASPARS, J.O.M. (1989)A Proposal for Dynamic Knowledge Representation. In: M. Penttone & E. Kuikka (eds.) *Poceedings of the Workshop on Alternatives of Logic Programming*, Kuopio University.

KONOLIGE, K. (1983) The data base as a model. *SRI Technical Note.* Menlo Park: SRI International.

MOORE, R.C. (1980) Reasoning about knowledge and action. *SRI Technical Note 191.* Menlo Park: SRI International.

ITK: P.O. BOX 90153 5000 LE TILBURG THE NETHERLANDS